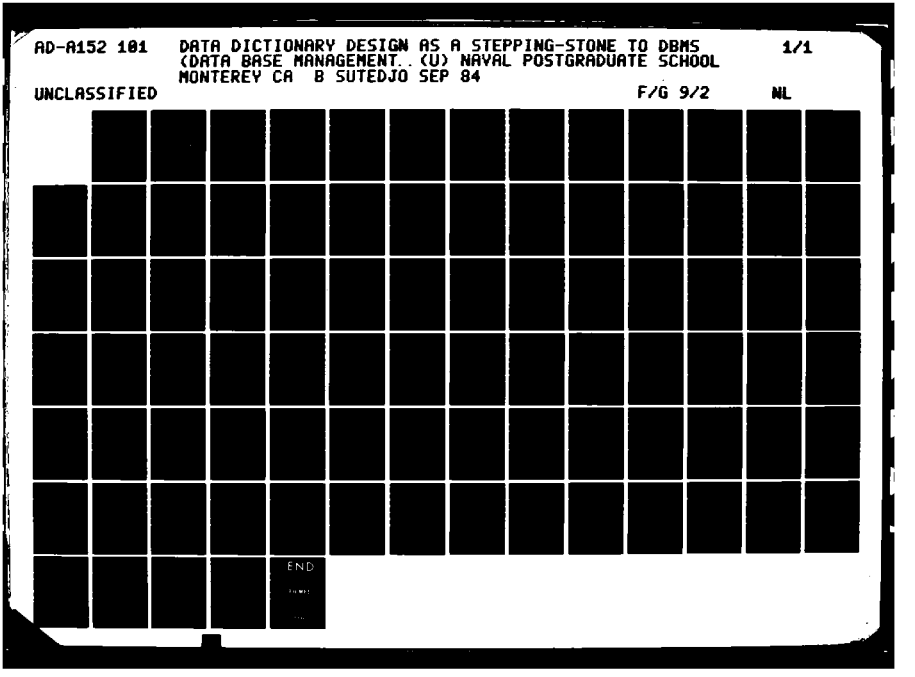
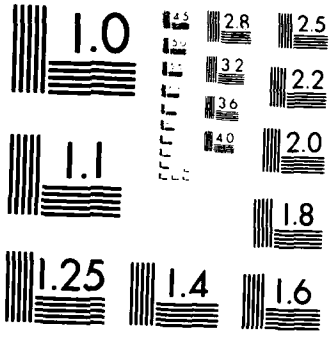

AD-A152 101
DATA DICTIONARY DESIGN AS A STEPPING-STONE TO DBMS (DATA BASE MANAGEMENT.. (U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA B SUTEDJO SEP 84
1/1
UNCLASSIFIED
F/G 9/2
NL
END



(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DATA DICTIONARY DESIGN
AS A STEPPING-STONE TO DBMS
IMPLEMENTATION IN THE INDONESIAN ARMY
DATA COLLECTING AND PROCESSING SERVICE

by

Bambang Sutedjo

September 1984

Thesis Advisor: Daniel R. Dolk

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A152 101	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Data Dictionary Design as a Stepping-stone to DBMS Implementation in the Indonesian Army Data Collecting and Processing Service		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bambang Sutedjo		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		12. REPORT DATE September 1984
		13. NUMBER OF PAGES 85
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Data Dictionary; DBMS Implementation; Indonesian Army; Data Dictionary Schema; DISPULLAHTAD; Data Dictionary Features; Entities; Relationships; Attributes

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The organizational structure, tasks, and system configuration of the Indonesian Army Data Collecting and Processing Service (DISPULLAHTAD) are presented briefly in order to provide a background for succeeding discussions. Features of available data dictionary systems (DDS) and the initial design of a data dictionary for current applications at DISPULLAHTAD are presented. Finally, based on that initial design, a recommendation for database management system (DBMS) implementation is discussed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

Data Dictionary Design as a Stepping-stone to DBMS Implementation in the Indonesian Army Data Collecting and Processing Service

by

Bambang Sutedjo
Captain, Indonesian Army
Armed Forces Military Academy, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September, 1984

Author: _____
Bambang Sutedjo

Approved by: _____
Daniel Roy Dolk, Thesis Advisor

_____
Norman R. Lyons, Co-Advisor

_____
Willis R. Greer, Jr., Chairman,
Department of Administrative Sciences

_____
Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

The organizational structure, tasks, and system configuration of the Indonesian Army Data Collecting and Processing Service (DISPULLAHTAD) are presented briefly in order to provide a background for succeeding discussions. Features of available data dictionary systems (DDS) and the initial design of a data dictionary for current applications at DISPULLAHTAD are presented. Finally, based on that initial design, a recommendation for database management system (DBMS) implementation is discussed.

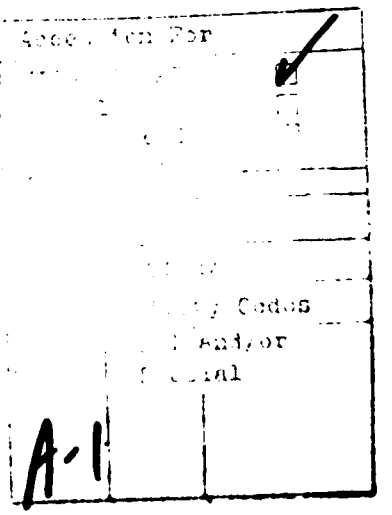


TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

As a young and developing information system organization, the Indonesian Army Data Collecting and Processing Service (DISPULLAHTAD) has a tremendous proliferation of application files. It is no surprise that there is much redundancy of data and efforts. Data redundancy wastes a limited resources, and furthermore, it raises the problem of inconsistent data, that is, the same element of data having different values within different files. The implementation of database management system (DBMS) could handle this problem by providing more control and more effective management of data.

On one hand, the information generated electronically becomes more and more in demand to the point where it has become a critical issue for the Indonesian Army. On the other hand, the personnel generating and maintaining this information move dynamically because of requirements for military tour of duty and tour of area. This situation creates problems in keeping accurate and up to date information. Even though the manual documentation is always done properly, this is not always adequate. It is often the case that many applications are highly dependent up on the personnel responsible for such applications. Standardized and centralized documentation is a "must", especially when a DBMS is implemented. In this regard, the data dictionary is a powerful vehicle that supports such documentation.

According to Dolk [Ref. 1], "a data dictionary is a collection of an enterprise's meta-data designed as one or more databases which can be retrieved and analyzed using standard database management system capabilities". This will be discussed further in a subsequent chapter, and will

be considered as a basis in choosing the most appropriate DBMS to be implemented by DISPULLAHTAD.

The organizational structure of DISPULLAHTAD, its system configuration, and its current various applications will be describes briefly in order to provide a background for the succeeding chapters. A discussion of database management system and a recommendation of the most appropriate DBMS to be implemented comprises the last chapter.

II. INDONESIAN ARMY DATA COLLECTING AND PROCESSING SERVICE (DISPULLAHTAD)

A. ORGANIZATION, TASK, AND SYSTEM CONFIGURATION

DISPULLAHTAD is an acronym in the Indonesian language that stands for the Indonesian Army Data Collecting and Processing Service. It was initiated in Fiscal Year 1973/1974 and formally organized in Fiscal Year 1975/1976 [Ref. 2]. DISPULLAHTAD is located in the Indonesian Army Headquarter - Jakarta, the capital city of the Republic of Indonesia.

The DISPULLAHTAD's main task is to provide all information processed electronically for all organizational elements of the Army requiring the information [Ref. 2]. In order to be able to accomplish this task, DISPULLAHTAD is equipped with several computer configurations. As of 1984, these include an IBM System 4341, an IBM System 370, several IBM System 3740s, and several TRS-80s.

In the lower organizational level such as Military Area Commands (KODAM), Army Finance Service (JANKUAD), Army Administrative and Personnel Service (JANMINPERSAD), Army Development and Educational Command (KOBANGDIKLAT), etc., each has its own Data Processing Service and it is called as PULLAHTA KOTAMA/LAKPUS or PULLAHTA for short. A simplified organizational structure of the Indonesian Army is presented as figure 2.1 and the position of both DISPULLAHTAD and PULLAHTAs can be clearly visualized.

Each PULLAHTA has two roles: first to process and provide all pertinent information requested by the organization to which it is attached, and secondly, to provide data

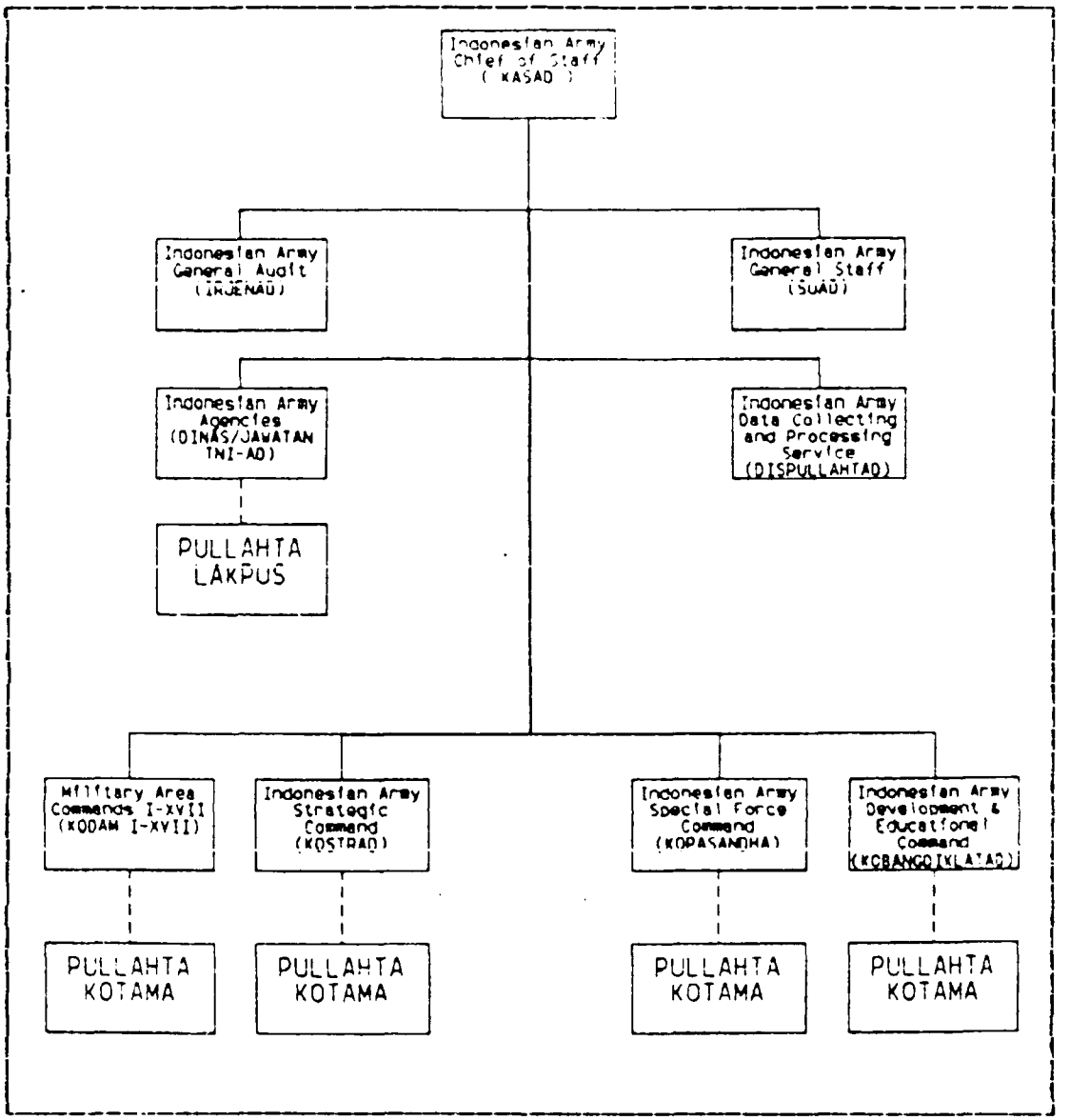


Figure 2.1 Indonesian Army Organizational Structure (Simplified Chart).

entry for all applications centrally processed by DISPULLAHTAD. In order to accomplish these objectives, each PULLAHTA is also equipped with some hardware as shown below.

1. PULLAHTAs inside Java Island

PULLAHTAs that belong to Military Area Command in Java island are equipped with an IBM System 4331 connected to the IBM System 4341 at DISPULLAHTAD in "online" mode via dedicated public telephone lines. This is the first stage in networking all PULLAHTAs throughout the country. The data interchange is done electronically through these dedicated lines.

2. PULLAHTAs outside Java Island

PULLAHTAs sited outside Java island are equipped with an IBM System 3740 and currently work in "off line" mode. Eventually they will be connected to DISPULLAHTAD via dedicated public telephone lines. The data interchange is done manually using floppy-disks transported via airline and it requires one to two days for the data to reach its destination.

B. APPLICATIONS

All applications done by DISPULLAHTAD are in order to fulfill its task of electronically providing information needed by the Indonesian Army. These applications are separated into three kinds of Management Information Systems [Ref. 3] :

1. Administration Management Information System

This category includes applications in finance, logistics, personnel, and all applications pertinent to development, education, and corps/specialty.

2. Military Management Information System

The applications of intelligence and security, territorial, communication and electronics, and organization, operation, and training are included in this category.

3. Planning and Controlling Management Information System

Three kinds of applications are included in this category: planning and budgeting, auditing and control, and command, control, and communication.

C. DATA REDUNDANCY PROBLEMS

There are many possible data redundancies within those applications. For instance consider data about name, rank, SSN, corps, occupation, etc., that belong to an individual assigned as Intelligence Officer in a Territorial Unit. His data will appear in at least four different files: personnel file, payroll file, intelligence file, and territorial file. If, for instance, there is a change to just one of those data elements, redundant effort is required to update all those four files and a high level of data inconsistency may result.

It happens many times that top level management detects data inconsistency in two different reports generated by DISPUILAHTAD (e.g. the total number of personnel appears differently in the personnel report and the payroll report). This has become very annoying, and tremendously reduces credibility in the computer system. In this regard, an effort should be made to eliminate the problem and one way of doing that is by designing and implementing a data dictionary system (DDS) in concert with a DBMS.

D. STAGED DEVELOPMENT APPROACH

The design of any information system is the most difficult and critical step. It should be done with great care and full awareness. As suggested by Sprague and Carlson [Ref. 4], there are three different approaches in initiating an information system:

• Quick-Hit approach.

This approach should be done if there is no clarification whether such an information system is needed or not, but there is a recognized high payoff for initiating the system. This approach requires developing the system in the most beneficial area, capturing the benefits, and then considering what to do next.

• Staged Development approach.

This approach is done by developing the system in the most beneficial area as in the quick-hit approach, but with some advanced and clear planning. Therefore, part of the effort in developing the first system can be reused in developing the second. This approach is very appropriate for initiating an information system clearly supported by top management, but with limited available resources.

• Complete System approach.

This approach requires the longest development time and highest development costs before any benefits are attained. Before building any part of the system, a full-service system generator and the organizational structure for managing it must be developed first. In this regard, this approach represents the most risky option.

For DISPULLAHTAD, it is anticipated that top management will strongly support the implementation of a DBMS, therefore the Quick-Hit approach is not necessary. On the other hand, limited computer resources makes a Complete System approach infeasible, too. Hence, the most appropriate approach is the Staged Development approach. In the Staged Development approach, identifying the functional area where there is a reason to expect the highest pay-off in starting the project is a crucial thing. Using the highest volume of transaction as the criteria, and by evaluating the transactional data gathered between April 1983 and December 1983

Month	Personnel Applic.	Payroll Applic.	Armament-Transp.	Building Applic.	Finance Reports	Planning Applic.	Others
Apr, 83	24,296	40,772	3,677	648	37,347	0	13,741
May, 83	123,527	36,969	2,852	7,446	81,253	0	6,281
Jun, 83	84,586	36,913	2,902	3,736	73,672	0	4,291
Jul, 83	22,045	28,782	3,542	0	23,176	0	1,815
Aug, 83	27,560	31,107	10,475	3,779	43,418	10,408	1,893
Sep, 83	32,080	38,978	3,190	0	24,713	1,080	0
Oct, 83	21,373	35,294	6,307	1,874	50,260	0	178
Nop, 83	34,202	37,983	6,473	13,177	30,427	0	8
Dec, 83	24,778	39,952	4,990	1,101	18,546	0	3,049
Total	394,447	326,750	44,408	31,761	382,812	11,488	31,256
Percentage	32.25	26.72	3.63	2.60	31.30	.94	2.56

Figure 2.2 Transactions by Application at DISPULLAHTAD.

(see Figure 2.2), there are three applications having high transactional volume: personnel, payroll, and finance report applications. Only the personnel and payroll applications have a master file which is maintained and used continucusly. Besides that, these two applications are the most crucial in maintaining personnel morale and the most often used in relation to the personnel management task.

Based on these evaluations, data used by the personnel and payroll applications will be the first database to be implemented by DISPULLAHTAD. These applications include personnel, payroll, intelligence personnel, and territorial personnel. It is also implied that the discussion on the DDS and DBMS will be limited to those applications.

The plan for this staged development approach are:

1. Initial design of DD covering personnel, payroll, intelligence personnel, and territorial personnel applications.

2. Implementation of personnel and payroll database.

3. Design of DD for all data used by the rest of applications excluded at the first stage.

4. Implementation of other databases.

III. DATA DICTIONARY

A. GENERAL

The revolutionary change in computer technology has created another challenge on how to organize and manage the very large-scale databases made possible by the combination of database management systems (DBMS) and powerful new hardware systems. The need to control the enterprise's data becomes critical due to the proliferation of microcomputers that trigger more and more applications which in turn creates redundancy and data inconsistency problems. At the same time, the number of microcomputer users demanding direct access to the enterprise data is also increasing. This direct access to large and complex databases again creates a problem of how to "coordinate" and control these complex information structures.

Data redundancy, data inconsistency, and the need to control the enterprise's data lead to the design and implementation of database systems. The database environment itself assumes an architectural plan designed to minimize redundancy and to emphasize accessibility. It assumes logical and physical structures aimed at separate objectives. It also assumes that individual file may serve many different applications. All of this is far too much complexity to be managed without precise and up-to-date documentation and control. The data dictionary is designed to define all appropriate aspects of the enterprise's data, so that it can be used as a tool to control and manage the database system no matter how great its size and how complex its structures.

B. INFORMATION RESOURCE MANAGEMENT (IRM)

The concept of IRM is that information is a vital enterprise asset that should be invested in, and used like other resources [Ref. 5].

IRM is the task of managing information resources such as data, processes, users, software, and hardware in an integrated and coordinated manner. IRM includes all management aspects of the information-related operations of an organization, such as policy formulation, resource allocation, implementation, and control.

A definition of IRM was formulated at a workshop on Data Dictionary Systems and Information Resource Management sponsored by the Association for Computing Machinery and the National Bureau of Standards in 1980:

> Information Resource Management is whatever policy, action, or procedure concerning information (both automated and non-automated) which management establishes that serve the overall current and future needs of the enterprise. Such policies, etc., would include considerations of availability, timeliness, accuracy, integrity, privacy, security, auditability, ownership, use, and cost-effectiveness.

This definition of IRM was chosen to emphasize the enterprise-wide nature of planning and execution of information policies, actions, and procedures in order that data can be treated as a true resource. It also reflects the primary shift of data processing uses from processing-centered design methodologies to data-centered methodologies.

1. Data Dictionary as the Tool of IRM

One of the problems encountered in IRM is the vast amount of data about information resources required to managed the enterprise data, together with the very complex and numerous relationships existing between them. This is

precisely the sort of task that a Data Dictionary System can be made to do, provided that it has been conditioned to know how to deal not just with data entities or process entities, but with the entire range of information resource entities.

2. Data Dictionary

A data dictionary is a collection of meta-data (data about the enterprise data) that could consist of: the name of data (including its synonyms and/or homonyms), the location of data, a description of the meaning of data, the relation between data, how the data is used, who is responsible for the data, the source of the data, etc., in short, a store of all the appropriate information about the data.

Recently, there has been a trend towards using data dictionary to include the following functions:

a. Definition of other data constructs such as records and files.

b. Definition of processes such as programs or manual processes.

c. Definition of data users whether individuals or organizational entities.

Along with these definitions, the data dictionary also began to be used to document the cross-references between them and to record their usage and organizational responsibilities.

3. Data Dictionary System (DDS)

A Data Dictionary System is a combination of software and procedures that aid an enterprise in setting up and maintaining its complex structure of data resources. The software itself may be produced in-house or acquired from software vendors. For the following three reasons, it is often better to purchase instead of building it in-house:

a. Design and Implementation

The task of designing and implementing a DDS, even one of modest functionality, is definitely a non-trivial one. There exists good potential that the magnitude of the task will be underestimated and that greater resources will be needed than those originally estimated.

b. Gaining a Success

If the use of DDS is to be at all successful, the software itself must conform to high standards of quality assurance. The use of the same software at many other installations, as is the case with a commercial package, aids in the early discovery of possible software errors and their corrections.

c. Technology Progress

There are good reasons for assuming that DDS technology will continue to progress and that substantial enhancements will significantly increase the usefulness and value of the DDS, and it will be difficult for an in-house system to keep pace.

There are several DDS commercially available now, such as DB/DC Data Dictionary of IBM, DATAMANAGER by MSP Inc., Integrated Data Dictionary (IDD) by Cullinet Software Inc., DATADICTIONARY by Applied Data Research Inc., Extended Data Dictionary (XDD) by Intell Systems Corp., UCC TEN by University Computing Company, and Data Control System (DCS) by Cincom Systems Inc.

In the following section, features that one could expect to find on those available DDS will be discussed. It must be pointed out that none of the available DDSs mentioned above will necessary have all explained features, and there is no such implication that all of these features are required in all DDS applications.

C. FEATURES OF A DATA DICTIONARY

There are several issues will be discussed here, such as Implementation and Architectural, Data Dictionary (DD) and DD Schema, Extensibility Facilities, Status Facilities, Dictionary Commands, Bridge Facilities, and Data Dictionary System Security.

1. Implementation and Architectural Issues

a. The Relationship Between DDS and DBMS

The primary purpose of a DBMS is to manage data, whereas the primary purpose of a DDS is to manage meta-data. Therefore, it is clear that there is a very little overlap between these two, in fact they are complementary; both functions are required for proper management of information resources.

Some of the functions a DDS will perform are in support of one or more DBMSs. This is to be expected as the DDS will manage all meta-data, including meta-data where the actual instances of data are stored in a database, which in turn is being managed by a DBMS. An element required for this latter function, the DBMS's management of data in databases, is the knowledge on the part of the DBMS of certain meta-data of the databases which are required by the DBMS in order for it to do its processing. This meta-data is commonly referred to as the **DBMS-Directory**, and it should be clear that this potentially is one area of overlap between the DDS and DBMS. In this sense, it is preferable to design a DDS prior to the DBMS implementation rather than to build a DDS that has to be fitted toward an existing DBMS. This reason together with an existing method of implementing a DDS as one of the DBMS's applications may explain why in this thesis designing the dictionary is done prior to the design of the databases.

b. The Method of DDS Implementation

It is preferable to design a data dictionary prior to the implementation of the DBMS. On the other hand, the implementation of that designed data dictionary as a complete DDS is a good candidate to be one of the DBMS applications, and indeed a number of existing DDS are implemented in this manner. But this is not a single option, the implementation of DDS can be either :

- DBMS-dependent system.

This is a DDS that uses a DBMS in its implementation.

- Free-standing system.

A DDS that doesn't use a DBMS in its implementation is considered to be included in this category.

There is no ultimate answer as to whether a free-standing or DBMS-dependent DDS is the best. There are both pros and cons to this question, and these depend on the enterprise's specialized circumstances, such as whether the enterprise implements a DBMS or not, and whether it uses multiple DBMSs or a single DBMS for its databases. Enterprise(s) that have no intention to implement a DBMS but DDS will give a favor toward a free-standing system. On the other hand, the enterprise(s) having multiple DBMS may implement a DBMS-dependent system and choose one of its DBMSs to implement it; or, they may implement a free-standing system in order to provide more flexible and fair control.

Other considerations include the DDS security and a view that the scope of DDS usage is substantially broader than the DBMS environment. With a DBMS-dependent system, personnel familiar with the use of the DBMS may find it easier to break the DDS security than would be the case with a free-standing system.

For the sake of completeness, there is another concept of DDS implementation method referred to as integrated DDS. This method offers an elimination of overlap between data dictionary and DBMS-Directory by combining these features into one. The advantage gained by combining these two features is that redundancy of storing the meta-data is eliminated.

c. Active and Passive DDS

In the processes that require meta-data for its execution, there should be a command or series of commands, representing some DDS functionality that produces the required meta-data. This functionality is called dictionary interface, and there are two kinds of such interface: active and passive interface.

An active interface means that all processes that require meta-data will use the most current meta-data in the data dictionary. Similarly, all processes which in the course of their execution generate meta-data are required to store the generated meta-data in the data dictionary.

On the other hand, the passive interface will have all that an active interface has to do as the option. The other option for all processes that require meta-data in its execution are either retrieve it from data dictionary or some other locations; or if the process already contains the meta-data, there exist an option for the system to check whether or not this meta-data is the most current version in the data dictionary. In the case of generating a meta-data, the process also has an option to store or not to store the generated meta-data.

Therefore, two conclusions can be drawn about the dictionary interface :

(1) A DDS may have some interfaces which are active and others which are passive

(2) The fact that an interface is active is a property not only of the DDS, but also of the overall system of which the DDS is a part.

2. Data Dictionary and Data Dictionary Schema

Data Dictionary denotes the organized and structured collection of meta-data which comprises the contents of the DDS. The data dictionary schema denotes the logical structure of the data dictionary, in a manner analogous to the use of same term in the context of a DBMS.

The structural characteristics of data dictionary and the contents of data dictionary schema will determine what kinds of meta-data can be stored in the data dictionary and what kinds of relationships can be established between them. Some systems have extensibility facilities whereby an installation can customize the data dictionary schema.

The schema is described in logical terms in order to gain a clearer insight about what kinds of meta-data are supported by the DDS. This logical description will, of course, be quite different from the manner in which these structures are actually implemented in a specific system. This description should be made independent of any implementation.

A data dictionary has a conceptual similarity to the **Entity-Relationship-Attribute** model. The basic unit in the data dictionary is a **Dictionary Entity** or **Entity** for short. Entities represent real world objects or things about which certain information exists in the data dictionary. And the information about entities themself, exists in the form of **Attributes** which generally denote the qualities or quantities of properties of the entities. Finally, data dictionary also contain information about **Relationships**

between entities, and relationships may have attributes assigned to them.

The term **Entity-type** is applied to some entities that have similarities among them. For example, if a set of files is described in the data dictionary, each file will be represented by a distinct entity. It then becomes useful to establish an Entity-type called File in the data dictionary and to say that all such entities representing files have the entity-type File. Attributes of entities of the same type will exhibit a certain degree of similarity. Entity-type File will likely have an attribute of what kind of access method used, and maybe another attribute showing the blocking factor used. These both access method and blocking factor are then called as an **Attribute-type** which is associated with the entity-type File. Beside the entity-type File, there will be an entity-type Record. The information would exist in the data dictionary explaining which types of records are included in a given file. All such relationships between these file entities and their associated record entities, then be called as a **Relationship-type.** In conclusion, the data dictionary schema would be viewed as containing all existing entity-types, relationship-types, and attribute-types. Any one of these three types may also be referred to as a **schema descriptor.**

Every entity in the data dictionary has a **primary name** which, depending on the particular DDS, will be unique either in the dictionary or within the entity-type to which the entity belongs. Some systems may facilitate duplicate user-supplied names by assigning them distinct sequential numbers. In this case, the concatenation of user-supplied name and sequential number constitutes the unique dictionary name. The allowable length of the primary name should be sufficiently large enough to convey the meaning of an entity in its primary name. It is common that at least some

entities will also be known by other names. Such alternate names are called **aliases** or **synonyms,** and the most important things are the capability of the DDS for tracking them and allowing access to the data dictionary via these alternate names. Sometimes it is convenient if non-unique synonyms are allowed. To fulfill this requirement, DDSs have facilities for tracking synonyms either as attributes of the respective entities, or as separate entities related to the primary entity. Therefore, it is important that the system should be able to recognize the context in which the synonym is used.

The attributes can be differentiated into some attribute-types, among which are: **Description** = it consists of an English language statement describing the meaning of the entity. **Classification keywords** = these are attached to the entities which then can be used for selective retrieval of these entities. **Audit-attributes** = these are attributes, generated by the DDS, indicating the identification of the person who created, the date of creation, the identification of last person who modified, the date of last modification, and the total numbers of modification, all for each entity.

The entity itself can be conveniently separated into three entity-types: **Data, Process,** and **Usage** entity-types.

a. Data Entity-types

The most common of this type, listed with typical attribute-types and relationship-types are:

(1) Item/Data Element. In some systems the lowest entity-type is **Item,** which is considered to be the atomic unit. In other systems, the lowest entity may be **Data Element,** which in its turn it may contain other Data Elements. This is usually specified by **contains** clause, which expresses the relationship.

Commonly provided attribute-types relate to the physical characteristics of the Item/Element, including distinctions between Source, Target, Internal representations, and the validation criteria that may be required for the real-world instances of the Item/Element.

(2) Group/Record. Systems that recognize the Item as an entity-type will contain **Group** as a separate entity-type, whereas systems that have Data Element as an entity-type do not have an entity-type for Groups. **Record** is logically the same as a Group, therefore separate entity-types for both of them may or may not exist. A relationship-type is provided to express the structure of the Group/Record. Commonly available attribute-types relate to the manner in which the constituent elements are aligned, and other physical characteristics of the entity.

(3) File. Relationship-types are provided to express the structure of the file. Attribute-types relate to the access method used, blocking and labelling information, etc.

(4) DBMS-related Entity-types. The entity-types that exist are dependent on the specific DBMS for which the DDS provides support services. In all cases, the entity-types equate to the various data descriptions used by the DBMS, such as Schema, Subschema, Database Directory, etc. Relationship-types and attribute-types are provided to allow the DDS to express the structure of these entities.

(5) Other Data Entity-types. Some DDSs offer Report, Screen, and Form entity-types. In each case, relationship-types are provided that allow the contents of such entities to be specified in terms of the constituent elements.

b. Process Entity-types

There are two most common Process Entity-types. They are **Program/Module** and **System/Subsystem** as will be discussed below.

(1) Program/Module. This entity-type represents information about a collection of executable code. Typical attribute-types are the language of the source code, the size, and the characteristics under which it operates. Relationship-types are provided to other Programs/Modules, as well as the data, i.e. databases, files, and elements, on which it operates. Generally speaking, different relationship-types are provided for input, output, and processing-in-place.

(2) System/Subsystem. This entity-type describes a collection of programs and/or Modules associated with a major function of the enterprise. Relationship-types are provided to associate a System with Subsystems, as well as the constituent Programs.

c. Usage Entity-types

Users and their organizational environment, and the **data communication environment** can be thought of as Usage (or External) Entity-types. They are not directly components of a system, such as data and processes, but nevertheless play an important role in its operation.

The User and organizational component entity-types may have relationship-types that allow users to be associated with organizational components. These components themselves, and selected relationship-types that associate users or components with data and process entities may describe the responsibilities assigned to those users.

The example of Data Communication environment entity-types are Terminals, Messages, and other entity-types

that describe the communication networks. Their relationship-types may provide the associations of such entities with other usage entities, i.e. a given terminal is assigned to a certain set of people or a certain organizational unit. Or it may provide such associations with data and process entities, i.e. a given terminal is authorized to execute only a given set of transaction programs, or to access only certain Files or Databases. It will be appropriate to note here, that the role of the DDS in these matters is strictly a repository for documentation, and that the DDS by itself **cannot** be expected to enforce such restrictions and limitations. In order for DDS to be used for enforcement, appropriate "active" interfaces would have to exist to assure that the restrictions and controls which are documented in the data dictionary are always invoked at execution time. It will, on the other hand, create more complexity and much overhead.

3. Extensibility Facilities

The concept of extensibility facilities is to allow an installation to modify the system-standard schema as delivered by the DDS vendor. Any new schema descriptor created through the use of extensibility facilities will be referred to as an **extensibility descriptor.**

Extensibility facilities are extremely powerful, and their usage should be done with great care because extensions to the system-standard schema, once they are used, can only be undone with some difficulties. Additionally, changes to the system may create confusion among the users of the DDS as well, and decrease their confidence in the system. Due to these reasons, it is recommended that their usage should be restricted to the **Dictionary Administrator,** the person who is responsible for DDS function, i.e., the recording of all meta-informations and meta-data and its

maintenance through the use of the DDS, along with making its facilities available to the users of the system.

There are three kinds of such facilities that exist in current DDSs :

a. Entity-type extensibility: the ability to add new entity-types to the dictionary.

b. Attribute-type extensibility: new attribute-types for either entity-types or relationship-types can be declared using this facility.

c. Relationship-types extensibility: this facility allows the installation to declare new relationship-types.

4. Status Facilities

These facilities allow the DDS to be used in a System Life Cycle environment where, for instance, a certain entity may both part of a production system and a new test system. Due to its intended usage, it is preferable to maintain the same name for the entity in different stages. Therefore, a facility is required that will allow two distinct dictionary entities to have the same name, yet different attributes and relationships. In some systems, such entities can be distinguished by assigning different version numbers to them. In systems having a status facility, it can be accomplished either by:

a. Appending the entity-status to the entity-name which provides the uniqueness of the name.

b. Logically partitioning the dictionary into separate databases for different statuses, and requiring uniqueness of the name only within each partition.

5. Dictionary Commands

A DDS may have one or more interfaces that allow a user to interact with the dictionary. Such an interface may in the form of :

- A command language.
- A screen-oriented interface.
- A fixed format batch data entity facility.
- A programmatic interface that allows user written application programs to access the dictionary.

A screen-oriented interface is more user-friendly compared to the others. It results in higher utilization of computer system resources, but makes the DDS available to a larger class of users. Another benefit may be that the error rate is substantially reduced.

The dictionary commands may be differentiated into eight categories on the basis of their functionality:

a. Dictionary Maintenance Commands: this enables an installation to create and maintain its data dictionary.

b. Reports and Queries Commands: an installation may generate reports using meta-data contained within the data dictionary using these commands.

c. Data Structure Interface Commands: enable other systems to use the descriptions of data structures contained within the dictionary.

d. Extensibility Commands: vehicle to exploit the extensibility facilities.

e. Status-related Commands: the ability to distinguish entities in different stages of the life cycle.

f. Security Commands: used to allow security declarations to be assigned to the dictionary.

g. Dictionary Processing Control Commands: used to control the dictionary process such as logon-logoff, processing defaults, etc.

h. Dictionary Administrator Commands: exclusive commands especially designed to be used by the dictionary administrator.

6. Bridge Facilities

Other facilities of a DDS exist in the form of bridges or interfaces to other systems. The contents of the dictionary may be made available as part of the processing functions of that system. In each case, these systems provide tools whose functionality is outside the DDS but which require data about entities which can be expected to exist in the dictionary. By accessing a dictionary and extracting the required information, the disadvantages of having to store and maintain redundant data is eliminated.

Those such interfaces are :

a. Report and Query System: the ability to support various kinds of reports and queries.

b. Validation Criteria: support other systems by providing a module which performs the specified validation and which can be inserted into a program.

c. Database Design: the ability to provide basic data needed by automatic database designers.

d. Test Data Generation: support test data generation by providing descriptions of the structures and formats of the files and databases.

7. DDS Security

As mentioned before, the term DDS security is applied here to denote the security of the DDS itself. Entities in the dictionary may have attributes describing access characteristics to the real-world instances of these entities, but this data is entirely informational in nature and cannot be enforced by the DDS, since the system is not part of the loop in the execution of programs against the "real data". On the other hand, unauthorized access to the DDS can be eliminated by applying such security procedures, e.g., the assignment of passwords, or the inclusion of

security levels which control the various kinds of access to dictionary entities. To gain more integrity and reliability, the security of the DDS must be considered to be related to the security of the entire computer system. The level of security existing in the computer system is influenced by the security of the basic systems software and the physical security of the installation, as well as the procedures used by the personnel of the installation. These latter are often notably lax, and it is not at all unusual to observe cases where passwords are not kept confidential and may, indeed, openly be shared with unauthorized people.

D. COST/BENEFIT ANALYSIS FOR DDS

As applied to any system acquisition, Cost/Benefit analysis should be done prior to and in order to get a justification for DDS acquisition, implementation, and usage. The following list of costs and savings represent tangible items that can be used in assessing the costs and benefits for an economic study of the feasibility of implementing a DDS.

1. Costs

There are eight possible costs which may be considered:

a. Acquisition cost is the accumulation of lease or purchase cost and the maintenance cost of the system.

b. Data Administration staff cost is self explanatory.

c. Hardware Cost is the sum of storage device cost and CPU time cost.

d. Start-up cost is the total cost of training data administrator staff and all activities such as developing a comprehensive plan (see Table I as an example) that should be done in any data dictionary implementation.

TABLE I

Comprehensive Plan for DDS's Usage

1. Development of a policy for the use of the DDS.
2. Development of standards to be followed in the dictionary, including naming conventions for dictionary entities.
3. Development of decisions on how to use the control facilities of the DDS, such as the status and security facilities.
4. Delegation of authority and responsibility for the use of various DDS facilities.
5. Definition of procedures for the use of the DDS, and development of the required policies to implement these procedures.
6. Design and implementation of customized features for the DDS, should any be required. This may include change to the dictionary schema to allow new types of information resources to be stored in the dictionary, production of specialized reports, or interfaces to other S/W systems.

e. Data collection cost is a function of the number of entities, attributes, and relationships which are to be put in the dictionary.

f. Maintenace cost will depend on the degree of changes to the application system or systems controlled by the DDS.

g. Application system change cost is a cost pertinent to any change to the application systems due to the implementation of the dictionary for reasons of efficiency, integrity, and maintainability.

h. User education cost is the cost for training people involved in data dictionary usage in addition to the data administrator staff.

2. Factors for Estimating Savings and Benefits

There are four factors which can be used as an aid in estimating the quantification of savings. The greater the degree to which these four factors are held to apply to the enterprise and its operations, the more the high end of estimated saving and benefits can be expected. Those four factors are:

a. The Maturity of an Information Processing Environment

This is a major factor in the benefits that can be attained with a DDS. Increased maturity will help substantially in the integration of dictionary facilities into the operations of the enterprise.

b. The Complexity of the Environment

The number of data elements, files, databases, and programs can be used to measure how complex the information processing environment is. Problems caused by complexity tend to worsen geometrically with the number of such elements, files, databases, and programs. The value of a DDS will be greater as complexity increases.

c. The Degree of Data Sharing

It should be common practice that data elements are shared by different programs, where some of these are from different systems. An important issue is that changes in one part of the system tend to have effects in many other parts of the system. Failure to compensate for such changes can cause production failures and unanticipated costs. Tracking the effect of these changes is a valuable feature of a DDS. It is made possible by evaluating the attributes and relationships of changed entities as the basis for further tracking.

d. Personnel Turnovers

The personnel associated with information processing systems is either data processing organization personnel or user's organization personnel used to deal with the information processing system. In this regard, a DDS offers two advantages: **First,** information which otherwise might be stored in the minds of individuals and which may be lost to the organization with the loss of the individual is now placed in the DDS. **Secondly,** the learning curve for new personnel is steeper than it would be without the use of a DDS.

3. Savings and Benefits

There are five areas in which savings and benefits may be expected. The four factors mentioned in the previous section can be used as aids in predicting specific monetary savings in each of these following areas:

a. System design and development: the prime advantage of the DDS is in its use for better communication between users and implementors. This results in fewer changes or iterations and consequently faster programming because the specification is better documented and understood by all parties.

b. System maintenance: better and more complete documentation in the dictionary, the ability to analyze the effect of proposed changes, and the improved communication between users and maintenance programmers on proposed changes or corrections.

c. Data redundancy: reduction of unplanned data redundancy will result in an improved system which has greater integrity and better operability as well as potentially decreased requirements for random storage devices.

d. Database creation: may take advantage of the descriptions contained within the DDS to reduce iterations in the design process and faster concurrence by all parties on the contents of a database.

e. Improved Communication: self explanatory.

IV. INITIAL DATA DICTIONARY DESIGN FOR DISPULLAHTAD

A. GENERAL

As mentioned before, the initial design of Data Dictionary will be limited to four application areas: personnel, payroll, intelligence personnel, and territorial personnel. This design is a first step in the Stage Development approach used, therefore it may be expanded in the future.

B. DATA DICTIONARY SCHEMA/SUBSCHEMA

In a manner analogous to the context of a DBMS, **Data Dictionary Schema** denotes the logical structure of the data dictionary (DD). In this regard, then, the term **Subschema** will denote a subset of the schema to be seen by a given application (process) or user [Ref. 6], and it is compatible with application views of a database [Ref. 7].

1. Data Dictionary Schema

The structural characteristics of the DD and the contents of the DD schema are important aspects of the usage of a DDS, since by evaluating these, users may know what kinds of meta-data and relationships between them exist within the DD. As suggested by Lefkovits et al [Ref. 8], the structural characteristics of a DD may be described in logical terms in order to gain a clearer insight of what kinds of meta-data are supported by the DDS. The logical structure of a typical DD drawn by Allen et al [Ref. 5] seem to be appropriate for DISPULLAHTAD's DD with only minor changes, and it is presented as figure 4.1.

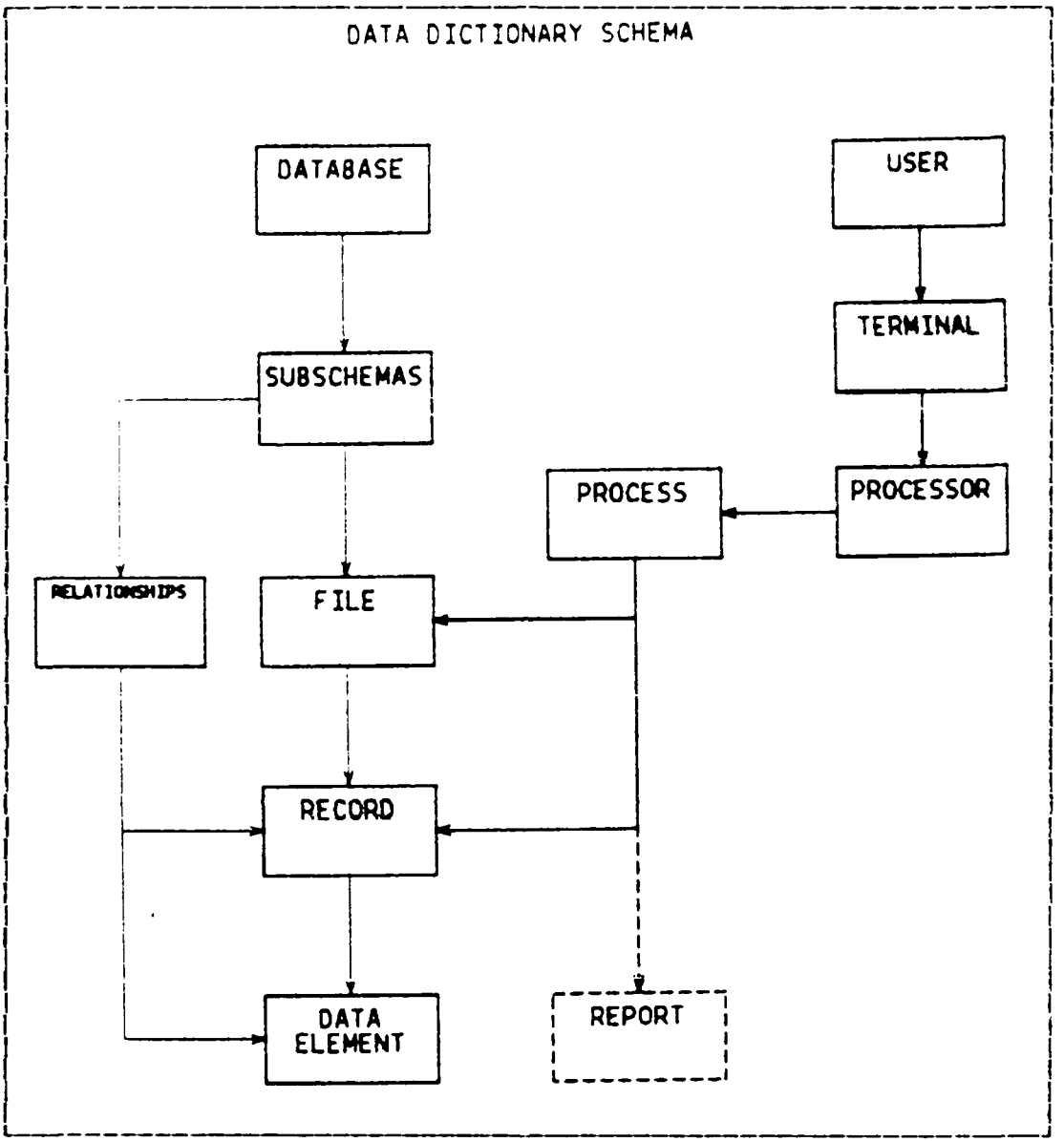


Figure 4.1 Logical Structure of DISPULLAHTAD's DD (adapted from Allen et al [Ref. 5]).

There are three kinds of entities (from the left to the right) within the logical DD in figure 4.1 :

a. Data Entities

These consist of database, subschema, relationship, file, group of elements, and data element entities. Wherein each record may consists of some elements, but may or may not have group of elements in it. File and/or record entity is the subject of a process entity, while data element entity is the subject of a transaction or a report entity.

b. Process Entities

A process entity may be an application program, a program module, or a system/subsystem that typically generates a report or does a transaction involving either data element or a group of data elements.

c. Usage Entities

Included in this category are user and their organizational environment (such as processors and terminals), and data communication environment (such as communication network, communication nodes, messages, etc.).

2. Data Dictionary Subschemas

Since there are four applications included in this initial design, there might be four subschemas accordingly. Due to the relatively small amount of meta-data that will be stored in the initial dictionary, however, and in order to reduce complexity, it is better not to apply subschemas at this point. Later, if the dictionary has grown substantially, and many applications have been added, the subschemas may be applied in order to improve efficiency and security.

C. DISPULLAHTAD'S DATA DICTIONARY

The design of this DD is based on current applications for which the DBMS has not been implemented yet. The following tables present the lists of Data, Process, and Usage entities and one or two of the actual instances of DISPUILAHTAD's data dictionary.

1. Entities

a. Data Entities

Table II summarizes data entities abstracted from the personnel application [Ref. 9 and Ref. 10]. Table III presents data entities for the payroll application [Ref. 11 and Ref. 12]. Data entities used in the intelligence personnel application is presented as Table IV [Ref. 13 and Ref. 14]. Finally, data entities from the territorial personnel application are shown in Table V [Ref. 15].

b. Process Entities

Process entities belonging to the four applications are presented as Table VI . This table lists only routine processes, not irregular processes, since the latter ones are not yet standardized (they are done as requested).

c. Usage Entities

Table VII presents user entities for the four applications.

TABLE II

Personnel Application Data Entities

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
I.	File	PERSFILE	File Personil	Personnel File
A.	Record	DAPOK	Data Pokok	Main Data Record
1.	Element	NOPERS	Nomor Personil	S.S.N. plus selfcheck
2.	Element	NAMA	Nama Personil	Personnel Name
3.	Element	PANGK	Pangkat/Golongan	Rank
4.	Element	CORPS	Corps/Cabang-Fungsi	Corps/Specialty
5.	Element	KOTAMA	Kotama/Lakpus	Unit of Command
6.	Element	SMKAL	Satminkal	Administrative Unit
7.	Element	KOKER	Kotama dan Satker	Logistics Unit
8.	Element	STPERS	Status Personil	Personnel Status
9.	Element	KTGORI	Kategori	Personnel Category
10.	Element	TGGORI	TMT Kategori	Date of Category
11.	Element	TGABRI	TMT ABRI/Pegawai	Date Joining the Service
12.	Element	TGLHR	Tanggal Lahir	Date of Birth
13.	Element	KLMIN	Jenis Kelamin	Sex
14.	Element	STWIN	Status Kawin	Marital Status
15.	Element	TGPJAB	TMT Prinlak Jabatan	Date of Implementing the Occupation
16.	Element	NOPRIN	Nomor Prinlak	Its Order Number
17.	Element	PRDARI	Prinlak Dari	Order From
18.	Element	JABAT	Jabatan	Occupation
19.	Element	STJABR	Penyangkatan dalam Jab.	Occupational Status
20.	Element	DIKUM	Pendidikan Umum Terakhir	Last General Education
21.	Element	TLAHIR	Tempat Lahir	Place of Birth
22.	Element	STARUM	Status Rumah Tinggal	Housing Status
23.	Element	KORDIN	Kode Rumah Dinas	House Code
24.	Element	SEPATU	Nomor Ukuran Sepatu	Shoes Number
25.	Element	TOPI	Nomor Ukuran Topi	Cap Number
26.	Element	TSHIRT	Nomor Ukuran T-Shirt	T-Shirt Number
27.	Element	TINGGI	Tinggi Badan	Height
28.	Element	BERAT	Berat Badan (Kg)	Weight (Kg)
29.	Element	WARNA	Warna Kulit	Skin Color

Table II

Personnel Application Data Entities (cont'd.)

30.	Element	RAMBUT	Rambut	Hair
31.	Element	DARAH	Golongan Darah	Blood
32.	Element	SETIA	Satya Lencana Kesetiaan	Medals
33.	Element	BAHDA	Bahasa Daerah	Local Language
34.	Element	BAHAS	Bahasa Asing	Foreign Language
35.	Element	AGAMA	Agama	Religion
36.	Element	TMTPA	TMT Perwira	Date of Promotion as an Officer
37.	Element	MDPFIK	Masa Dinas Pa Fiktif	Fiction Svc Years
38.	Element	MKFIK	Masa Kerja Yang Diperhitungkan	Fact Service Years
39.	Element	SUKU	Sukubangsa / Keturunan	Tribe
40.	Element	KLASIF	Klasifikasi Psychotest	Psychotest Classification
41.	Element	SEMAPTA	Kesamaptaan	Physical Fitness Test
42.	Element	NDPP	Nomer DPP	Payroll Number
43.	Element	ANAKL	Jumlah Anak Laki	Number of Sons
44.	Element	ANAKP	Jumlah Anak Perempuan	Number of Daughters
45.	Element	SARKAP	Kesatuan Kaporlap	Unit Giving Uniform
46.	Element	SATBOV	Kesatuan Ransum	Unit Giving Ration
47.	Element	SATCB	Kesatuan Dosir Personil	Unit of Administration
48.	Element	SATGAS	Kesatuan Penugasan	Unit of Duty
49.	Element	NOIND	Nomor Induk	Civilian S.S.N.
50.	Element	MINNO	Minno Jubar Subjubar	Unit of Payroll
51.	Element	DIKMIL	Dik Militer Tertinggi	Highest Mil. Education
52.	Element	KLASKES	Klasifikasi Kesehatan	Health Classification
53.	Element	PUANFIS	Kemampuan Fisik	Physical Ability
54.	Element	PUANSUS	Kemampuan Khusus	Special Ability
E.	Record	JABATAN	Jabatan	Occupation Record
1.	Element	TGJAB	TMT Jabatan	Date of Occupation
2.	Element	KOJAB	Kode Jabatan	Occupation Code
3.	Element	PKTJAB	Pangkat Saat Itu	Rank at the Beginning of the Occupation
4.	Element	NOKJAB	Nomor Skep Jabatan	Occupational Order Number
5.	Element	KDRJAB	SKEP Dari	Order From
6.	Element	STJAB	Pengangkatan Dalam Jab.	Occupation Status
7.	Element	IAKJAB	Pelaksanaan Jabatan	Job Implementation

Table II

Personnel Application Data Entities (cont'd.)

C.	Record	URAJAB	Uraian Jabatan	Occupational History Rec.
1.	Element	TGJABU	TMT Jabatan	Date of Occupation
2.	Element	PKTJABU	Pangkat Saat Itu	Rank at the Beginning of Occupation
3.	Element	NOKJABU	Nomor Skep Jabatan	Occupation Order Number
4.	Element	KDRJABU	SKEP Dari	Order From
5.	Element	JABATU	Jabatan	Occupation, Description
6.	Element	STJABU	Pengangkatan Dalam Jab.	Occupation Status
7.	Element	IAKTABU	Pelaksanaan Jabatan	Implementation
D.	Record	PANGKAT	Pangkat/Golongan	Rank/Category Record
1.	Element	PKAT	Kode Pangkat	Rank Code
2.	Element	TGKAT	TMT Pangkat	Date of the Rank
3.	Element	KEPKAT	Nomor Skep Pangkat	Number of Promotion Order
4.	Element	KDRKAT	Skep Pangkat Dari	Promotion Order From
5.	Element	MKKAT	Macam Kenaikan Pangkat	Kind of Promotion
E.	Record	RDIKMIL	Pendidikan Militer	Military Education Record
1.	Element	TGDIK	Tanggal Mulai Pendidikan	Date of Education
2.	Element	KODIK	Kode Pendidikan	Education Code
3.	Element	IOKDIK	Tempat Pendidikan	Place of Education
4.	Element	HSLDIK	Hasil Pendidikan	Result of Education
5.	Element	IAMDIK	Lama Pendidikan	Duration of Education
6.	Element	JMLSIS	Jumlah Siswa	Number of Student
7.	Element	NORUT	Nomor Urut Lulus	Graduated as Number
F.	Record	CPERASI	Operasi	Operational Record
1.	Element	TGOPS	Tanggal Mulai Operasi	Date of Operation
2.	Element	MAOPS	Macam Operasi	Kind of Operation
3.	Element	NAMOPS	Nama Operasi	Name of Operation
4.	Element	IOKOPS	Lokasi Operasi	Place of Operation
5.	Element	KEPOPS	Sprin Operasi	Operation Order
6.	Element	KEPSAT	Sprin Dari	Order From
7.	Element	SELOPS	Tanggal Selesai Operasi	Date Operation Finished

Table II

Personnel Application Data Entities (cont'd.)

8.	Element	KORBAN	Korban	Victims
9.	Element	TJASA	Tanda Jasa	Operation Medals
G.	Record	KLUARGA	Keluarga	Family Record
1.	Element	HUBKEL	Hubungan Keluarga	Family Relationship
2.	Element	TGLHRK	Tanggal Lahir Keluarga	Date of Birth
3.	Element	NAMAK	Nama Keluarga	Name
4.	Element	TPLHRK	Tempat Lahir Keluarga	Place of Birth
5.	Element	KLMINK	Kelamin Keluarga	Sex
6.	Element	KERSTA	Pekerjaan / Status	Occupation
7.	Element	TGSTA	Tanggal Berlakunya Status	Date of Occupation
8.	Element	AGAMAK	Agama Keluarga	Religion
H.	Record	KETRKEL	Keterangan Keluarga	Detail of Family Record
1.	Element	DIKKEL	Pendidikan Keluarga	Family Member's Education
2.	Element	TGKAW	Tanggal Kawin	Date of Marriage
3.	Element	TOKKAW	Tempat Kawin	Place of Marriage
4.	Element	ALAMAT	Alamat Lengakp	Address
I.	Record	ADRES	Alamat Personil Bujangan	Bachelor Address Record
1.	Element	ALAMAT	Alamat	Personnel Address
J.	Record	PENGLN	Pengalaman Luarnegeri	Foreign Experience Record
1.	Element	TGBRK	Tanggal Berangkat	Date of Departure
2.	Element	NEGARA	Kode Negara Tujuan	Code of Foreign Country
3.	Element	MTUGAS	Macam Tugas	Kind of Duty
4.	Element	UTUGAS	Uraian Tugas	Detail of Duty
5.	Element	TGKEMB	Tanggal Kembali	Date of Return
K.	Record	PGARAN	Pelanggaran	Breach Record
1.	Element	TGGAR	Waktu Kejadian	Date of Breach
2.	Element	MAGAR	Macam Pelanggaran	Kind of Breach
3.	Element	TPGAR	Tempat Kejadian	Place of Breach

Table II

Personnel Application Data Entities (cont'd.)

	4.	Element	CARA	Cara Yang Digunakan	Method of Breach
	5.	Element	MOTIV	Motivasi Pelanggaran	Motivation of Breach
	6.	Element	AKGAR	Akibat Kejadian	Result/Effect of Breach
	7.	Element	TDKDAN	Tindakan Dan Sat	CO action
	8.	Element	HKPOK	Hukuman Pokok	Main Punishment
	9.	Element	BESHK	Besarnya Hukuman Pokok	Total of Punishment
	10.	Element	HKTAM	Hukuman Tambahan	Additional Punishment
	11.	Element	TGDHK	Waktu Dijatuhkannya Huk.	Date of Punishment
	12.	Element	TPMHK	Tempat Pelaksanaan Huku.	Place of Punishment
M.		Record	RDIKUM	Pendidikan Umum	General Education Record
	1.	Element	TALULS	Tahun Selesai/Lulus	Date of Graduation
	2.	Element	NDIKUM	Nama Pendidikan	Name of Education
	3.	Element	GELAR	Gelar/Keterangan	Title of Education
		File	PERWIRA	Personil Perwira	Officer Personnel File
A.		Record	SUMBPA	Sumber Perwira	Source of Officer
	1.	Element	THNPA	Tahun Lulus Sebagai Pa	Graduation Date as Off.
	2.	Element	KEYPA	Nama Personil	Personnel Name
	3.	Element	NOPRS	Nomer Personil	SSN plus Selfcheck Digit
B.		Record	DAPEN	Dapen Perwira	Officer Conduct Record
	1.	Element	THNMIL	Tahun Penilaian dan Periode ke	Evaluation Semester
	2.	Element	NILAI	Nilai	Grade
	3.	Element	KLAPEN	Klasifikasi	Classification
C.		Record	PSYCHO	Psychotest	Psychotest Record
	1.	Element	THNPSY	Tahun Test	Year of the Test
	2.	Element	KLAPSY	Klasifikasi	Classification

TABLE III

Payroll Application Data Entities

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
	File	FILEDPP	File DPP	Payroll File
A.	Record	DAPOKDPP	Data Pokok	Main Data Record
1.	Element	KTM	Ku Kotama	Finance Command
2.	Element	NAP	Pekas/NA	Unit of Payroll
3.	Element	JBR	Juru Bayar	Finance Officer
4.	Element	SJB	Sub Jubar	Disbursing Number
5.	Element	SAT	Kesatuan/Satker	Unit of Command
6.	Element	KDP	Kelompok DPP	Payroll Group
7.	Element	SPH	Status Penghasilan	Payroll Status
8.	Element	NRP	NRP/NBI/ NIP	S.S.N.
9.	Element	NAM	Nama	Personnel Name
10.	Element	PKT	Pangkat/Golongan	Rank
11.	Element	TGL	Tanggal Lahir	Date of Birth
12.	Element	MKG	Masa Kerja Gaji	Payroll Service Year
13.	Element	TMT	TMT MKG	Date of Service
14.	Element	GPK	Gaji Pokok	Main Salary
15.	Element	SKL	Status Keluarga	Marital Status
16.	Element	JUM	Jumlah Anak	Number of Children
17.	Element	KTS	Kategori Skorsing	Schorsing Category
18.	Element	BLN	Bulan Warakawuri	Widow No. of Month
19.	Element	KOJ	Pot Persekot Gaji	Payroll Discount
20.	Element	GRG	Pot Ganti Rugi	Indemnification Discount
21.	Element	SWR	Pot Sewa Rumah	Housing Discount
22.	Element	STO	Keastuan Organik	Unit of Duty
23.	Element	SRC	Status Record	Record Status
24.	Element	TKU	Tunjangan Khusus	Special Allowance
25.	Element	SUL	Status ULP	Ration Status
26.	Element	PTT	Pangkat Tituler	Tituler Rank
27.	Element	NTM	Nomer Tambahan	Selfcheck Digit
28.	Element	KLM	Kode Kelamin	Sex Code
29.	Element	DPP	Nomer DPP	Payroll Number

Table III

Payroll Application Data Entities (cont'd.)

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
B.	Record	LAPMUT	Laporan Mutasi	Transaction Record
1.	Element	KTM	Ku Kotama	Finance Command
2.	Element	NAP	Pekas/NA	Unit of Payroll
3.	Element	JBR	Juru Bayar	Finance Officer
4.	Element	SJB	Sub Jubar	Disbursing Number
5.	Element	SAT	Kesatuan/Satker	Unit of Command
6.	Element	NRP	NRP/NBI/ NIP	S.S.N.
7.	Element	NTM	Nomor Tambahan	Selfcheck Digit
8.	Element	NAM	Nama	Personnel Name
9.	Element	PKT	Pangkat/Golongan	Rank
10.	Element	DPP	Nomer DPP	Payroll Number
11.	Element	JMT	Jenis Mutasi	Transaction Code
12.	Element	TMT	TMT Mutasi	Date of Transaction
13.	Element	KBR	Keadaan Baru	Post Transaction Data
14.	Element	KDP	Kelompok DPP	Payroll Group
15.	Element	MCM	Macam Perubahan	Transaction Category

TABLE IV

Intelligence Personnel Application Data Entities

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
	File	FILINTEL	File Personil Intel	Intell. Pers. File
A.	Record	PERSINTL	Record Personil Intel	Intell. Pers. Rec.
1.	Element	NAMA	Nama	Name
2.	Element	PANGKAT	Pangkat	Rank
3.	Element	CORPS	Corps	Corps/Specialty
4.	Element	NOPERS	Nomer Personil	S.S.N. plus Selfcheck
5.	Element	JABATAN	Jabatan	Occupation
6.	Element	SATMINKL	Satminkal/Kesatuan	Unit of Command
7.	Element	KOTAMA	Kotama/Lakpus	Command
8.	Element	ALAMAT	Alamat Rumah/Keluarga	Home Address
9.	Element	TPTLAHIR	Tempat Lahir	Place of Birth
10.	Element	TGLAHIR	Tanggal Lahir	Date of Birth
11.	Element	AGAMA	Agama	Religion
12.	Element	ISTRI	Nama Istri/Suami	Spouse Name
13.	Element	ANAKLK	Jumlah Anak Laki-laki	Number of Son
14.	Element	ANAKPR	Jumlah Anak Perempuan	Number of Daughter
15.	Element	DIKUM	Pendidikan Umum Tertinggi	General Education
16.	Element	DIKMIL	Pendidikan Mil. Tertinggi	Military Education
17.	Element	DIKJUR	Pendidikan Kejuruan (MOS) Non-Intel	Special Training Non-Intelligence
18.	Group	DIKINDN	Pendidikan Intel Dalam Negeri	Intelligence Course in Indonesia
19.	Element	MADIKDN	Macam Pendidikan	Course Name
20.	Element	LAMDIKDN	Lama Pendidikan (Minggu)	Course Duration in Week
21.	Element	LULUSDN	Kode Lulus/Tidak Lulus	Code of Graduated or Not
22.	Element	TAHUNDN	Tahun Pendidikan	Year of Course
23.	Element	TEMPATDN	Tempat Pendidikan	Course Location
24.	Group	DIKINLN	Pendidikan Intel Luar Negeri	Intelligence Course in Foreign Country
25.	Element	TEMPATLN	Tempat Pendidikan (Neg.)	Course Location (Country)
26.	Element	LULUSLN	Lulus/Tidak Lulus	Code of Graduated or Not
27.	Element	TAHUNLN	Tahun Lulus	Year of Graduation

Table IV

Intelligence Personnel Application Data Entities (cont'd.)

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
28.	Group	BAHASING	Kemampuan Bahasa Asing	Foreign Language Ability
29.	Element	ASING	Bahasa Asing	Name of Foreign Language
30.	Element	AKPASING	Aktif/Pasif	Active or Passive
31.	Group	BAHADAER	Kemampuan Bahasa Daerah	Local Language Ability
32.	Element	DAERAH	Bahasa Daerah	Name of Local Language
33.	Element	AKPDARAH	Aktif/Pasif	Active or Passive
34.	Group	RIWAJAB	Riwayat Jabatan	History of Occupations
35.	Element	NOKEP	Nomer Surat Keputusan	Order Number
36.	Element	TANGGAL	Tanggal	Date of Order
37.	Element	TMTJAB	TMT Jabatan	Date of Occupation
38.	Element	JABATAN	Jabatan	Occupation
39.	Element	PANGKAT	Pangkat	Rank

TABLE V

Territorial Personnel Application Data Entities

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
	File	FILETERR	File Personil Terr.	Terr. Pers. File
A.	Record	PERSTERR	Record Personil Terr	Terr. Pers. Record
1.	Element	ESELON	Eselon	Organization Level Code
2.	Element	KODAM	Kodam	Military Area Command
3.	Element	KOREM	Korem	Military Resort Command
4.	Element	KODIM	Kodim	Military District Command
5.	Element	KORAMIL	Koramil	Military Rayon Command
6.	Element	JABATAN	Kode Jabatan	Occupation Code
7.	Element	NAMA	Nama	Name
8.	Element	PANGKAT	Pangkat	Rank
9.	Element	CORPS	Corps	Corps/Specialty
10.	Element	NRP	NRP/NBI	S.S.N.
11.	Element	TAHIR	Tanggal Lahir	Date of Birth
12.	Element	DIKMIL	Kode Pendidikan Militer	Military Education Code
13.	Element	DIKTER	Kode Pendidikan Ter	Terr. Education Code
14.	Element	TARTER	Kode Penataran Ter	Terr. Course Code
15.	Element	TMTJAB	TMT Jabatan	Date of Occupation
16.	Element	PENGAL	Pengalaman Dalam Apter	Terr. Experiences
18.	Element	PDA	Kode PDA	Service Continuation Code
19.	Element	MPP	TMT MPP	Date of Resignation
20.	Element	SENJATA	Kode Senjata Organisasi	Armament Code
21.	Element	RAN	Kode Kendaraan Org.	Transportation Code
22.	Element	RUMAH	Kode Rumah Tinggal	Housing Code
23.	Element	SANGGUP	Kode Kesanggupan Kerja	Duty Ability Code
24.	Element	ANAK	Jumlah Anak	Number of Children
25.	Element	ISTERI	Nama Isteri	Spouse Name

Table V

Territorial Personnel Application Data Entities (cont'd.)

No.	Entity Type	Entity Name	Description (Indonesian)	Description (English)
B.	Record	MUTTER	Laporan Mutasi	Transaction Record
1.	Element	KOREM	Korem	Military Resort Command
2.	Element	KODIM	Kodim	Military District Command
3.	Element	KORAMIL	Koramil	Military Rayon Command
4.	Element	NRP	NRP/NBI	S.S.N.
5.	Element	MUTASI	Kode Mutasi	Transaction Code
6.	Element	FIELD1	Field Pertama	First Field
7.	Element	FIELD2	Field Kedua	Second Field
8.	Element	FIELD3	Field Ketiga	Third Field
9.	Element	KODE	Kode Pers/Mat	Personnel/Material Code

TABLE VI

Process Entities

No.	Pertinent To Appl.	Entity Name	Description (Indonesian)	Description (English)
1.	Personnel	ELIGIBLE	Membuat daftar personil yang eligible untuk kenaikan pangkat (tiap semester)	List all personnel eligible for rank promotion (each semester)
2.	Personnel	FEEDBACK	Cetak data personil perorangan, untuk feed-back ke kesatuan (tiap semester)	Print all individual personnel data as feed-back to the lowest unit level (each semester)
3.	Personnel	MUTASI	Update Master File Personil (tiap bulan)	Personnel Master File Up-Dating (monthly)
4.	Payroll	DPPR01	Konversi MF DPP Bulan Ini dari Versi Lama ke Versi Baru (tiap bulan)	Conversion of Current Payroll Master File from Old to New Version (monthly)
5.	Payroll	DPP01A	Edit Laporan Mutasi di Pullahta Kotama (tiap bulan)	Payroll Transaction Data Editing in Pullahta Kotama (monthly)
6.	Payroll	DPPS01	Edit Laporan Mutasi di Dispullahtad (tiap bulan)	Payroll Transaction Data Editing in Dispullahtad (monthly)
7.	Payroll	DPPS02	Sort Laporan Mutasi by No. DPP (tiap bulan)	Payroll Transaction Data Sorted By Payroll Number (monthly)
8.	Payroll	DPPS03	Sort Master File DPP Bulan Lalu by No. DPP (tiap bulan)	Old Payroll Master File Sorted by Payroll Number (monthly)

Table VI

Process Entities (cont'd.)

9.	Payroll	DPPS04	Pemutakhiran Master File DPP (tiap bulan)	Payroll Master File Up-Dating (monthly)
10.	Payroll	DPP05A	Edit Data Pokok di Pullahta Kotama (tiap bulan)	Payroll Main Data's Data Editing in Pullahta Kotama (monthly)
11.	Payroll	DPPS05	Edit Data Pokok di Dispullahtad (tiap bulan)	Payroll Main Data's Data Editing in Dispullahtad (monthly)
12.	Payroll	DPPS06	Sort Data Pokok by No. DPP (tiap bulan)	Payroll Main Data's Data Sorted By Payroll Number (monthly)
13.	Payroll	DPPS07	Check Double Data Pokok (tiap bulan)	Double Check of Payroll Main Data (monthly)
14.	Payroll	DPPS08	Pemberian No. DPP (tiap bulan)	Create Payroll Number to Main Data (monthly)
15.	Payroll	DPPS09	Sort Merge antara Dapok Bernomer DPP dengan Dapok Bernomer DPP Baru di-create (tiap bulan)	Sort Merge by Payroll Number between Payroll Main Data with Number and with New Number (monthly)
16.	Payroll	DPPS10	Matching antara New Master I dengan Gabungan Dapok (tiap bulan)	Matching between New Master File and Merged Main Data (monthly)
17.	Payroll	DPPS11	Sort Merge antara New Master II dengan Matched Dapok (tiap bulan)	Sort Merge by Payroll Number between Newest Master File with Matched Main Data (monthly)

Table VI

Process Entities (cont'd.)

18.	Payroll	DPPS12	Check DPP double (tiap bulan)	Check for double Main Data (monthly)
19.	Payroll	DPPS13	Perbaikan DPP double (tiap bulan)	Verifying the doubled Main Data (monthly)
20.	Payroll	DPPS14	Sort Master by NRP (tiap bulan)	Master File Sorted by SSN (monthly)
21.	Payroll	DPPS15	Create Notam (tiap bulan)	Calculate SSN's selcheck (monthly)
22.	Payroll	DPPS16	Pemisahan Master per Kotama/Lakpus (tiap bulan)	Separate Master File by Kotama/Lakpus (monthly)
23.	Payroll	DPPS17	Sort Master per Kotama by NRP (tiap bulan)	Sort Separated Master File by SSN (monthly)
24.	Payroll	DPPS18	Create Master Hitung per Kotama/Lakpus (tiap bulan)	Create Master for Payroll Lists for each Kotama (monthly)
25.	Payroll	DPPS19	Cetak DPP (KU-107, KU-108, Rekap Pot. Persit KCK) (tiap bulan)	Print Payroll Lists (monthly)
26.	Payroll	DPPS20	Sort/Merge Master per Kotama untuk Cetak Kekuatan Pangan (tiap bulan)	Sort/Merge to combine the Separated Master File for Ration Report (monthly)
27.	Payroll	DPPS21	Cetak KU-102 (tiap bulan)	Print Ration Report (monthly)

Table VI

Process Entities (cont'd.)

28.	Payroll	DPPS22	Sort/Merge Laporan Mutasi Ditolak (tiap bulan)	Sort/Merge to combine Bad Transaction Data (monthly)
29.	Payroll	DPPS23	Cetak Laporan Mutasi Ditolak untuk Feed-back (tiap bulan)	Print Bad Transaction Data for Feed back (monthly)
30.	Payroll	DPPS24	Sort/Merge Dapok Ditolak (tiap bulan)	Sort/Merge to combine Bad Main Data (monthly)
29.	Payroll	DPPS25	Cetak Dapok Ditolak untuk Feed-back (tiap bulan)	Print Bad Main Data for Feed back (monthly)
30.	Intell.	UPDATE	Updating Data Perorangan Apintel (tiap saat)	Updating the Intelligence Personnel Record (any time)
31.	Intell.	FEEDBACK	Cetak Data Perorangan Apintel untuk feed-back (tiap 6 bulan)	Print the Intelligence Personnel Record for Feed-back (each semester)
32.	Terr.	UPDATE	Updating Data Perorangan Apter (tiap saat)	Updating the Territorial Personnel Record (any time)

TABLE VII

User Entities

No.	Entity Name	Description (Indonesian)	Description (English)
1	DISPUL	Dinas Pengumpulan dan Pengolahan Data TNI-AD (DISPULLAHTAD)	Indonesian Army Data Collecting and Processing Service (DISPULLAHTAD)
2	SPAM	Staf Pengamanan TNI-AD (SPAM SUAD)	Indonesian Army Intelligence Staff (SPAM SUAD)
3	SPERS	Staf Personil TNI-AD (SPERS SUAD)	Indonesian Army Personnel Staff (SPERS SUAD)
4	STER	Staf Teritorial TNI-AD (STER SUAD)	Indonesian Army Territorial Staff (STER SUAD)
5	JANKUAD	Jawatan Keuangan TNI-AD (JANKUAD)	Indonesian Army Finance Service (JANKUAD)
6	MINPERS	Jawatan Administrasi dan Personil TNI-AD (JANMINPERSAD)	Indonesian Army Administrative and Personnel Service (JANMINPERSAD)
7	DALKAR	Dinas Pengendalian Karier TNI-AD (DISDALKARAD)	Indonesian Army Carrier Control (DISDALKARAD)
8	UMUM	Semua eselon organisasi pada TNI-AD	General, all user entities in Indonesian Army Organization

2. Relationships

The relationships can be represented by the following relations:

RELATIONSHIPS(relation_name,key,composite_key,secondary_key,
other_entity_names)
---- key ----

a. Relation Name

In pre-DBMS situation, this may be filled with the record entity name to represent the relationships between element data entities. In the case where a DBMS has been implemented, it should be filled with the relation name since a logical record may consist of some relations in order to reduce complexity and/or fulfill the five normal forms.

b. Key

This is an entity name that is used as the key for both storing and accessing the relation. If the relation uses composite keys, this will be the first part of the composite keys where the second part will be stored as composite-key attribute.

c. Composite Key

This is the entity name used as the second part of a composite key. If primary key is not composite, this attribute will be filled with "NONE".

d. Secondary Key

This is the entity name used as the secondary key. If the relation has no secondary key, this attribute will be filled with "NONE".

e. Other Entity Names

The names of other entities (except keys) in the relation will be filled by this attribute. An ampersand sign (&) will be used between two entity names and a sentence of "REPETITIONS OF" will be written in front of a repeating group of elements.

3. Attributes

Every entity has information attached to it called attributes. In the following, all information that may be included as attributes will be discussed.

a. Data Entities

There will be several pieces of information included for each data entity (either file, record, or element entity). Since the DD typically be implemented as one of the DBMS application, these can be represented as a relation of:

```
FILE_ENTITY(entity_name,block_size,access_method,logical_
                                record_size,physical_storage_device)
            --- key ---
RECORD_ENTITY(entity_name,length,fixed_variable_code,key,
                          composite_key,secondary_key,
                          updating_time,updating_mode)
              --- key ---
ELEMENT_ENTITY(entity_name,length,code,source,user,
                           definition)
               --- key ---
```

(1) Entity Name.

The name of the entity is limited to a maximum of eight characters, this may be a combination of alphabetic and numeric, but should contain an alphabetic as its first character.

(2) Block Size.

Self explanatory.

(3) Access Method.

This may be encoded as:

S - Sequential

I - Indexed Sequential

D - Direct Access

V - Virtual Storage (VSAM)

(4) Logical Record Size.

This is equal to record length.

(5) Physical Storage Device.

This may be coded as the following:

TAPE - Magnetic Tape

DISK - Magnetic Disk

DRUM - Magnetic Drum

FICP - Floppy Disk / Diskette

CARD - Punched Card

PAPR - Paper Tape

(6) Fixed / Variable Code.

The codes used are:

FIX - Fixed Length Record

VAR - Variable Length Record

(7) Key.

This is an entity name used as the key for both storing and accessing the relation. If the relation uses composite keys, this will be the first part of the composite keys where the second part will be stored as composite-key attribute.

(8) Composite Key.

This is the entity name used as the second part of composite key. If a primary key is not composite, this attribute will be filled with "NONE".

(9) Secondary Key.

This is the entity name used as the secondary key. If the record has no secondary key, this attribute will be filled with "NONE".

(10) Updating Time.

This attribute contains a description about how often this record will be updated. This will be a number of days.

(11) Updating Mode.

This attribute contains description about how the updating is done. It is encoded as follows:

BATCH - Batch Processing
ONLIN - Online Processing
BOTH - Both of Batch and Online

(12) Entity Length.

This denotes the length of an entity (may be record, or element). In the case of variable record, this information will be filled with zeroes, since the length of each record will be attached within the record itself.

(13) Entity Code.

This is a code for the character type:

A - Alphabetic
N - Numeric
AN - Aphanumeric.

(14) Source.

This denotes the organizational entity responsible for providing, updating, and deleting the entity.

(15) Users.

This denotes the organizational entity (entities) allowed to retrieve and use the entity. If subschemas are applied to the DD, this attribute will not be necessary.

(16) Definition.

This provides a detailed narrative describing the entity. This may include the information about frequency_of_update, range_of_acceptable values, and so on.

(17) Alias.

At the present time, every entity name (especially element entity) has no alias attribute attached to. In the future, this attribute sure will be needed. One way to accomodate this need is by add another relation called alias relation in which has at least two attributes: entity-name and its alias-name.

b. Process Entities

The information included as attributes in the process entity are: name, input_entity, output_entity, and the description of the process. These can be represented as the relation of:

```
PROCESS(process_name,input_entity,output_entity,description_
                     of_process,descr_of_input,input_media,
                     descr_of_output,output_media)
        ---- key ---
```

But, since a process may have more than one of either input or output entity, this relation should have a composite keys rather than having only the name as its single key. Because only one input/output entity is allowed in every instance of PROCESS [Ref. 6], processes having more than one input/output entity may waste storage since all attributes will appear unnecessarily more than once. In the case that this relation has a composite key, a query asking which data entities are input (or output) to a given process also possesses a difficulty since this relation can not be retrieved using only the process_name (it must be retrieved using its composite key, instead). In order to make a better payoff, those attributes may be arranged using the following three relations:

```
PROCESS(process_name,description)
        ---- key ---

PROCESS_INPUT(process_name,input_entity,description,
                                        media)
              ---------- key ----------
```

PROCESS_OUTPUT(process_name,output_entity,description,
media)
---------- key ----------

(1) Process Name.

Process name will be limited to maximum of eight characters as applied to data entities.

(2) Description of Process.

Self explanatory.

(3) Input Data.

The input data may be either a file, record, group of elements, data element, or data entered from console.

(4) Description of Input.

This attribute may be filled with a description such as the input data is "sorted by RANK" for instance.

(5) Input Media.

As to physical storage device attribute, this attribute may be filled with input storage device, or data entered via console.

TAPE - Magnetic Tape
DISK - Magnetic Disk
DRUM - Magnetic Drum
FLCP - Floppy Disk / Diskette
CARD - Punched Card
PAPR - Paper Tape
CONS - Data Entered via Console

(6) Output Data.

The output data may be magnetic-data, displayed data, or printed material.

(7) Description of Output.

This attribute may be filled with a description such as the output data is "sorted by KOTAMA" for instance.

(8) Input Media.

This attribute may be filled with output storage device or printed output material and these are encoded as follows:

TAPE - Magnetic Tape
DISK - Magnetic Disk
DRUM - Magnetic Drum
FIOP - Floppy Disk / Diskette
CARD - Punched Card
PAPR - Paper Tape
PRIN - Printed Material

c. Usage Entities

The information pertinent to these entities have been included and can be derived from data entities in terms of who is responsible for update operations and who is allowed to retrieve a data entity. Here, this information will be stated again from the reverse point of view, that is, which data are the responsibility of this entity, and which data are allowed to be retrieved by this entity. Information that will be included as the usage entity's attributes are: user_name, description of the user, entity_name, and type_of_access. For reasons similar to those discussed concerning the process entities, since a given user may have more than one data-entity as its responsibility and/or to be retrieved to, these attributes may be represented by the following three relations:

USER(user_name,description)
-- key --

USER_ACCESS(user_name,entity_name,type_of_access)
-------- key --------

USER_RESPONSIBILITY(user_name,entity_name)
-------- key --------

(1) User Name.

As with the data entity and process entity names, the usage entity name will be limited to a maximum of eight characters, toc.

(2) Entity Name.

Entity_name here means a data_entity_name.

(3) User Responsibility.

This may be a list of file, record, group of element, or data element entities that are this user's responsibility. This can be viewed as subschema.

(4) User Access.

This may be a list of file, record, group of element, or data element entities that may be retrieve by this user. This can be viewed as subschema, and may or may not be same as the list of user responsibility items.

(5) Type of Access.

The type_of_access is either:

R - Read only

U - Update only

B - Both R and U

N - No access

(6) Description of Entity.

Self explanatory.

d. Summary of Relations

Table VIII summarizes the relations of relationships, data entities, process entities, and usage entities.

e. Example of Relations

Table IX presents examples of file, record, and element entity relations. Examples of process entity relations and user entity relations are presented as Table X. Finally, an instance of the relationship relation is shown in Table XI.

TABLE VIII

Relational Model for DISPULLAHTAD Dictionary System

```
REIATICNSHIPS:
        RELATIONSHIPS(relation_name,key,composite_key,secondary_key,
                                                    other_entity_name)
                      ---- key ----
DATA-ENTITIES:
        FILE_ENTITY(entity_name,block_size,access_method,logical_
                                                record_size,physical_storage_device)
                    --- key ---
        RECORD_ENTITY(entity_name,length,fixed_variable_code,key,
                                                composite_key,secondary_key,updating_
                                                time,updating_mode)
                      --- key ---
        ELEMENT_ENTITY(entity_name,length,code,source,user,definition)
                       --- key ---
PROCESS-ENTITIES:
        PROCESS(process_name,description)
                ---- key ---
        PROCESS_INPUT(process_name,input_entity,description.media)
                      ---------- key ----------
        PROCESS_OUTPUT(process_name,output_entity,description,media)
                       ---------- key ----------
USAGE-ENTITIES:
        USER(user_name,description)
             -- key --
        USER_ACCESS(user_name,entity_name,type_of_access)
                    -------- key --------
        USER_RESPONSIBILITY(user_name,entity_name)
                            -------- key --------
```

TABLE IX

Examples of Data Entity Relations

Element Entities.

Relation:
ELEMENT-ENTITY(entity_name,length,code,source,user,definition)
--- key ---

Entity-name	Entity length	Char. type code	User-respons. (source)	User-allow to access	Definition
NOPERS	14	AN	MINPERS	UMUM	SSN plus selfcheck-digit and duplication number
NAMA	26	A	MINPERS	UMUM	Personnel name, birth certificate is the only formal source document

Record entities.

Relation:
RECORD-ENTITY(entity_name,length,fixed_variable_code,key,composite_key,
secondary_key,updating_time,updating_mode)
--- key ---

Entity-name	Record length	Fixed/variable code	Primary-key	Composite key	Secondary key	Updating Time	Uplating Mode
PERSINTL	295	F	NOPERS	NONE	NONE	1	ONLIN
DAFCKDPP	128	F	DPP	NONE	NONE	30	BATCH

Table IX

Examples of Data Entity Relations (cont'd.)

File Entities.

Relation:
FILE-ENTITY(entity_name,block_size,access_method,logical_record_size,
physical_storage_device)

--- key ---

Entity-name	Block-size	Access-method	Logical-record-size	Physical-storage-device
FILEINTL	10	V	295	DISK
FILETERR	20	V	128	DISK

TABLE X

Examples of Process & User Entity Relations

Process Entities.

Relation:
PROCESS-ENTITY(process_name,description)
--- key ----
PROCESS-INPUT(process_name,input_entity,description,media)
--------- key ----------
PROCESS-OUTPUT(process_name,output_entity,description,media)
---------- key ----------

Process name	Process description	Input name	Input description	Input media	Output name	Output description	Output media
DPPS11	Sort merge by Payroll Numb. between New Masterfile & Matched Data	DAPOKDPP	Matched Main Data (output of DPPS10)	FLOP	DAPOKDPP	New Master-file ready as the in-put to DPPS12	TAPE
		DAPOKDPP	New Master-file pre-merge (output of DPPS10)	TAPE			

Table X

Examples of Process & User Entity Relations (cont'd.)

User entities.

Relation:
USER_ENTITY(user_name,description)
-- key --
USER_ACCESS(user_name,entity_name,type_of_access)
-------- key --------
USER_RESPOSIBILITY(user_name,entity_name)
-------- key --------

User entity	Description of User entity	Allowed access to entities	Type of access	Responsible for entities
SPAM	Indonesian Army Intelligence Staff (SPAM TNI-AD)	PERSFILE PERSINTL PERSTERR	R B R	PERSINTL
SPERS	Indonesian Army Personnel Staff (SPERS TNI-AD)	PERSFILE PERSINTL PERSTERR	B R R	PERSFILE

TABLE XI

Examples of Relationship Relation

Relationship Entities.

Relation:
RELATIONSHIPS(relation_name,key,composite_key,secondary_key,other_entity_names)
---- key ----

Relation name	Primary key	Composite key	Secondary key	Other Entity Names
PERSINTL	NOPERS	NONE	NONE	NAMA & PANGKAT & CORPS & NOPERS & JABATAN & SATMINKL & KOTAMA & ALAMAT & TPTLAHIR & TGLAHIR & AGAMA & ISTRI & ANAKLK & ANAKPR & DIKUM & DIKMIL & DIKJUR & REPETITIONS OF : (MADIKDN & LAMDIKDN & LULUSDN & TAHUNDN & TEMPATDN) & REPETITIONS OF : (TEMPATLN & LULUSLN & TAHUNLN) & REPETITIONS OF : (ASING & AKPASING) & REPETITIONS OF : (DAERAH & AKPDARAH) & REPETITIONS OF : (NOKEP & TANGGAL & TMTJAB & JABATAN & PANGKAT)

4. Example of Data Dictionary's Queries

Meta-data contained within data dictionary may be used to answer questions asked by top managers, users, or technical staff (such as system analysts, programmers, and operators). In the following, several queries and the corresponding responses will be presented.

a. Top Management Queries

Top management may ask a question like: "How often is the personnel masterfile updated ?"

Possible answer is:

PERSFILE is updated once every 30 days.

b. User Queries

The user responsible for personnel management may ask the following question: "I need a list of personnel having rank of captain who speak French fluently and are experienced in the intelligence field. Is DISPULLAHTAD able to provide these data ?".

Possible answer is:

See entities:

1. PANGKAT (rank) in relation PERSINTL
2. ASING (foreign language) in relation PERSINTL
3. AKPASING (active/passive code) in relation PERSINTL

c. Technical Staff Queries

"What are the inputs and/or outputs of process DPPS17 ?", is one possible question asked by an operator for instance.

Possible answer is:

Process DPPS17

Input is/are:

1. DAPOKDPP (sorted by DPP), media: TAPE

Output is/are:

1. DAPOKDPP (sorted by KOTAMA), media: TAPE
2. KEKPANGN (sorted by KOTAMA), media: DISK

d. Unanticipated Queries

One of the advantages of designing the dictionary using the relational model is it can accommodate unanticipated queries. As one of the DBMS's application, this dictionary supports any query expressible in a relaticnal query language (e.g. SEQUEL).

V. THE IMPLEMENTATION OF DATABASE AT DISPULLAHTAD

A. DATA DICTIONARY DESIGN AS A STEPPING-STONE

The implementation of a database at DISPULLAHTAD may take advantage of the design of Data Dictionary in the preceding chapter in many ways:

- The designed DD may be used as the first DBMS application. Then, the experience gained here can be used in the future application of the real database implementation.
- DD as a repository of all meta-data will provide a full specification and description of all entities and relationships between them. Given this, the implementation of DBMS will be faster due to fewer changes and iterations in database development.
- In the case where an automatic database design tool is used, such as DATA DESIGNER, it may be interfaced with the DDS in order to take an advantage of the DDS contents. The database designer may benefit from the full descriptions of each entity contained within the data dictionary.

In this regard, the design of DD for current DISPULLAHTAD applications can be considered as a stepping-stone to the DBMS implementation.

B. DISPULLAHTAD'S DATABASE DESIGN

From Tables II through V, it can be seen that there are many data element redundancies within the four applications. These redundancies need to be eliminated by designing a database fulfilling the five normal forms. One possible method is by gathering the common data elements in one relation, and other specific data pertinent to each of the four applications in separate relations. The specific data model

for each application must likely consist of more than one relation.

In designing the database, availability of data description may be exploited to make this work easier. For example, suppose the following relationship is designed in the personnel database:

MAINPERS (nopers, nama, pangkat, corps, jabatan, satminkl)
--key-

Here, the possible descriptions contained within the dictionary that may be extracted are:

- Which files and records would have to be accessed in order to establish an instance of this relation ?
- What is/are the key/composite keys of each records derived from the preceding query ?
- What is the length of each of those entities ?

Furthermore, in order to satisfy the five normal forms a full and clear description of each entity is needed. These descriptions are contained within the data dictionary. The issue of actual database design is beyond the scope of this thesis and is left for possible follow-on thesis work.

C. CHOOSING THE DATA DICTIONARY SYSTEM (DDS)

1. Features

The available commercial DDS have most of the following features (see Figure 5.2 for more detail).

a. Dictionary Schema

This is a feature used to generate a manufacturer's standard schema, such as entities, relationships, and attributes.

b. Schema Extensibility

This is a feature whereby an installation is able to customize the manufacturer's standard schema by adding to it new entity-types, relationship-types, and attribute-types.

c. Dictionary Maintenance

This is a feature that enables an installation to create and maintain its data dictionary.

d. Reports and Queries

An installation may generate reports using meta-data contained within the data dictionary. A DDS provides these abilities via these features.

e. Bridge/Interface Facility

This feature generates descriptions from the data dictionary needed by other systems, typically an application development tool such as DATA DESIGNER.

f. Program Access Facility

This enables an installation to extend the functionality of a DDS, i.e., the preparation of programs able to access data dictionary contents.

g. Status Facility

This feature provides the status of each entity, especially when the data dictionary is used in the system life cycle, by providing aids in application development.

h. Security Facility

An installation may restrict access to the data dictionary to authorized personnel only. This requirement is fulfilled by this feature.

2. DISPULLAHTAD's DDS Requirements

a. Data Dictionary Features

There are five data dictionary features required in order to implement DISPULLAHTAD's data dictionary. These are:

- **Dictionary Schema** which is needed in order to generate the entities, relationships, and attributes.
- **Schema Extensibility** which will be needed to accommodate specific needs that cannot be fulfilled by the dictionary schema.
- **Dictionary Maintenance** which is used to create and maintain the data dictionary.
- **Reports and Queries** which are required in order to generate reports from and to extract data contained within data dictionary.
- **Security Facility,** even though the data dictionary has no actual instances of "secured" data, it will contain information about such data that can be used to access it, i.e., entity_name, access_method, where such data are stored, etc. Therefore, a security facility is required to add and strengten the level of security.

Other features are optional. These can be considered as "nice to have".

b. Active Versus Passive Data Dictionary

A full-active or partially-active system is very desirable because it provides features such as enforcing standards, range_of_value auditing, transaction monitoring, etc. On the other hand, an active system possesses much overhead, suffers in terms of longer turn-around time, and requires more complex processing algorithms. Therefore, at this time, a **passive data dictionary system** is more appropriate for DISPULLAHTAD. In the future, after enough

experience has been gained, a partial-active system may be applied in order to take fuller advantage of the data dictionary.

c. Free-standing Versus DBMS-dependent System

A free-standing system is very appropriate for an installation having different DBMSs. This may happen in an installation with databases using network or hierarchical structures in conjuction with newer technology such as the relational system. In this case, all systems may access the data dictionary independently, since the usage of the data dictionary is not limited to any one system.

On the other hand, a data dictionary may be implemented as one of the DBMS applications (some DDSs are implemented in this manner) [Ref. 8]. This approach is appropriate for an installation implementing databases using a single DBMS, and DISPULLAHTAD falls into this category. Therefore, a DBMS-dependent system will provide more advantages for DISPULLAHTAD, e.g., it can be used as a training tool in implementing the database. Another possible advantage is that if DISPULLAHTAD should change from a passive to an active system, there will not be too many modifications required because the data dictionary and the databases are already compatible.

d. Make or Buy

Buying an available commercial system may provide a high quality and ready-to-use system. Figure 5.2 summarizes features of current commercial systems that may be used in choosing the best system fulfilling the required features. A commercial system used by many installations without much trouble may be an indication that it has a certain quality. Criteria listed in Figure 5.1 may be used in selecting the best system for DISPULLAHTAD.

1. It should have at least the following five features that may be considered as the primary criteria:

   a. Dictionary Schema.
   b. Schema Extensibility.
   c. Dictionary Maintenance.
   d. Reports and Queries.
   e. Security Facility.

2. The following four criteria may be considered as the secondary criteria:

   a. Compatible with current hardware.
   b. Compatible with applied DBMS.
   c. High quality assurance.
   d. Low Acquisition and Set-up Costs.

Figure 5.1 Criteria for Choosing Commercial Data Dictionary System.

On the other hand, one significant advantage of designing a DDS in-house is that the system can be fitted to specific requirement of the installation. Furthermore, given that DISPULLAHTAD will implement a DBMS, the relational dictionary presented in the previous chapter may be a good candidate for the first application. By implementing the dictionary as a relational database, only one system needs to be acquired (DBMS) instead of two (DBMS and DDS).

3. Recommendation

For the reasons discussed in the previous section, it is better for DISPULLAHTAD to implement the data dictionary model described in the previous chapter as the first application of the DBMS. The dictionary will be a **DBMS-dependent system.** Initially, the system should be **passive,** and later, if appropriate, it may be changed to an active system.

Features :	DB/DC (IBM)	DATAMANAGER (MSP, Inc.)	IDD (Cullinet)	DATADICTIONARY (Applied-Data Rsch)	XDD (Intell)	UCC TEN	DCS (CINCOM)
Dicticnary-Schema	*	*	*	*	*	*	*
Schema-Extens.	*	*	*	*	*	N/A	N/A
Dicticnary-Maint.	*	*	*	*	*	*	*
Peports & Queries	*	*	*	*	*	*	*
Security-Facility	*	*	*	*	*	*	*
Bridge/ Interface	*	*	*	*	*	*	*
Status-Facility	*	*	*	*	*	*	N/A
Prcgram Access Facility	*	N/A	N/A	N/A	N/A	N/A	N/A
Command Stream and Perform Command	N/A	*	N/A	N/A	N/A	N/A	N/A

Figure 5.2 Features of Current Commercial DDS (summarized from Lefkovits et al [Ref. 1]).

VI. CONCLUSION

Implementing a DBMS is a "must" for DISPULLAHTAD in order to control the proliferation of its applications that in turn raises problems of data redundancy and data inconsistency. Primarily, a DBMS provides data manipulation capabilities whereas a data dictionary provides management and control. Applying management and control (by implementing DDS) first will make the job of database design and implementation easier in term of lessening the difficulties and the time and effort required to develop databases. In this regard, designing a data dictionary may be considered as a stepping-stone to the implementation of a DBMS.

This thesis has presented a relational model of a dictionary which can satisfy the needs of DISPULLAHTAD. The advantages of this model are:

1) it is compatible with any relational DBMS that DISPULLAHTAD may procure.

2) it obviates the need to buy a DDS in addition to the DBMS.

3) it can be tailored specifically to DISPULLAHTAD's needs because of the flexibility of the relational model.

4) it satisfies the criteria for a DDS (see Figure 5.1)

In order to attain the objective of managing and controlling data resources, the following implementation policy is recommended:

- All personnel involved in software development (such as system analysts, programmers, etc.) should use the data dictionary extensively in doing their jobs.
- Only the data administrator staff may update the data dictionary, others may access it in read-only mode.

• All suggestions concerning the data dictionary may be addressed to the data administrator staff.

This thesis has stopped short of suggesting a database design for DISPULLAHTAD's personnel application. Follow-on work could be done using the dictionary model suggested herein as a foundation.

LIST OF REFERENCES

1. Dolk, Daniel Roy, A Dictionary/Directory System (DDS) for SPLICE, Naval Postgraduate School, 1984

2. Indonesian Army HQ, Memorandum on the Project of Indonesian Army Data Collecting and Processing Service, dated 5 May 1976, the Office of Assistant of Planning, Research, and Development to the Army Chief of Staff, Indonesian Army HQ, 1976

3. Indonesian Army Data Collecting and Processing Service, Indonesian Army Management Information Systems - Draft, undated, Indonesian Army Data Collecting and Processing Service

4. Sprague Jr, Ralp H., and Eric D. Carlson, Building Effective Decision Support Systems, Prentice-Hall, Inc., 1982

5. Allen, Frank W., Mary E.S. Loomis, and Michael V. Mannino, The Integrated Dictionary/Directory System, Computing Surveys, Vol. 14, No. 2, June 1982

6. Kroenke, David M., Database Processing: Fundamentals, Design, Implementation, Second Edition, Science Research Associates, Inc., 1983

7. Goldfine, Alan H., Editor Database Directions Information Resource Management - Strategies and Tools, NBS Special Publication 500-92, ACM, National Bureau of Standards, Issued September 1982

8. Lefkovits, Henry C., Edgar H. Sibley, and Sandra L. Lefkovits, Information Resource/Data Dictionary Systems, QED Information Sciences, Inc., 1983

9. Indonesian Army HQ, Indonesian Army Chief of Staff Instruction, No.:INS/7/VII/1980, dated 17 July 1980, Indonesian Army HQ, 1980

10. Indonesian Army HQ, Indonesian Army Chief of Staff Instruction, 1st Amendment No.:INS/7a/VII/1980, dated 17 February 1983, Indonesian Army HQ, 1980

11. Indonesian Army HQ, Technics Directive of Payroll Application, No.:JUKNIS/09/IX/1981, dated 29 September 1981, Indonesian Army HQ, 1981

12. Indonesian Army HQ, Technics Directive of Payroll Application, 1st Amendment, No.:JUKNIS/09a/IX/1981, dated 5 January 1983, Indonesian Army HQ, 1983

13. Indonesian Army HQ, Technics Directive of Intelligent Personnel Application, No.:JUKNIS/057/VI/1978, dated 15 June 1978, Indonesian Army HQ, 1978

14. Indonesian Army HQ, Technics Directive of Intelligent Personnel Application, No.:JUKLAK/16/IX/1982, dated 23 September 1982, Indonesian Army HQ, 1982

15. Indonesian Army HQ, Technics Directive of Territorial Personnel Application, Indonesian Army HQ, 1976

INITIAL DISTRIBUTION LIST

No.		Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, Califcrnia 93943	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
3.	Department Chairman, Code 54 Department of Administrative Science Naval Postgraduate School	1
4.	Computer Technology Department, Code 37 Naval Postgraduate School Monterey, Califcrnia 93943	1
5.	Assistant Professor Daniel Roy Dolk, Code 54DK Department of Administrative Science Naval Postgraduate School	1
6.	Asscciate Professor Norman R. Lyons, Code 54LB Department of Administrative Science Naval Postgraduate School	1
7.	Bambang Sutedjo, Kapten Chb Nrp 26701 Dispullahta TNI-AD, Jl. Veteran No. 5 Jakarta Pusat, INDONESIA	1
8.	Library of Dispullahtad Dispullahta TNI-AD, Jl. Veteran No. 5 Jakarta Pusat, INDONESIA	2
9.	Tony A Rumbayan, Kolonel Czi Dispullahta TNI-AD, Jl. Veteran No. 5 Jakarta Pusat, INDONESIA	1
10.	Djoko Sardjono, Kolonel Cpl Dispullahta TNI-AD, Jl. Veteran No. 5 Jakarta Pusat, INDONESIA	1
11.	Triyono, Letnan Kolonel Czi Dispullahta TNI-AD, Jl. Veteran No. 5 Jakarta Pusat, INDONESIA	1
12.	Dadat Kustiwa, Letnan Kolonel Chb Dispullahta TNI-AD, Jl. Veteran No. 5 Jakarta Pusat, INDONESIA	1
13.	Syafei Djamil, Letnan Kolonel Chb 1121 Seventh Street Monterey, CA 93940	1
14.	Moch. Subekti, Kapten Laut (E) 1121 Seventh Street Monterey, CA 93940	1
15.	Widhya B Prawiraatmadja, Kapten Inf 1121 Seventh Street Monterey, CA 93940	1

END

FILMED

5-85

DTIC